Available online at www.sciencedirect.com

ScienceDirect

Journal of Electrical Systems and Information Technology 2 (2015) 207–218

www.elsevier.com/locate/jesit

A neural network-based approach for recognizing multi-font printed English characters

Najmeh Samadiani *, Hamid Hassanpour

Shahrood University, Department of Computer Engineering and Information Technology, Daneshgah Street, Shahrood, Iran

Available online 30 June 2015

Abstract

In this paper, we propose a method for recognizing English characters in different fonts. The proposed method based on neural network is resistant to font variant. When the samples in new fonts are added to the database, the accuracy of existing methods rapidly decreases and they are not resistant to font variant but to the accuracy of proposed method that almost stays constant and does not much decrease. A similarity measure neural network is used to identify characters and similarity measure compares the features of characters and the features of the indicators associated with the characters from A to Z obtained in the training stage. We use similarity measure instead of distance measure in SOM neural network because a person learns font-independent and a literate can read without knowing the font of the written note. In fact he/she measures similarity between the notes in new fonts and learned notes in his/her mind. Therefore, we use two samples for training the network as representative of all fonts such as default notes in man's mind. We could obtain 98.56% accuracy of recognizing a database that includes 24 different fonts in 11 different sizes.

Keywords: Character recognition; Similarity measure; Feature extraction; SOM neural network

1. Introduction

Optical character recognition (OCR) has been an active research area for many scholars, because this technology is widely applied to the car license plate recognition, barcode recognition, sorting of postal letters automatically and many other areas of application (Yang et al., 2011). OCR is used for recognizing both printed and handwritten letters and many researches have been presented to make recognition with higher performance and in a faster manner.

There are some common methods to recognize printed and handwritten characters. Rahman and Fairhurst (1998) exploited multiple-expert classification to provide new approaches to the processing of printed data. Four well-known handwritten character recognition algorithms (binary weighted scheme (BWS), frequency weighted scheme (FWS), moment-based pattern classifiers (MPC), multilayer perceptron and back propagation (MLP)) are used to recognize printed characters on specific databases containing limited fonts and accuracy of 97.16% is achieved. A system is

* Corresponding author. Tel.: +98 9151871285.
E-mail address: najmeh_sam@yahoo.com (N. Samadiani).
Peer review under the responsibility of Electronics Research Institute (ERI).

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jesit.2015.06.003

presented for identification and recognition of handwritten and typewritten text from document images using hidden Markov models (HMMs) by Huaigu et al. (2011). It is shown that the contextual constraints from the HMM significantly improve the identification performance over the conventional Gaussian mixture model (GMM)-based method. Type identification is then used to estimate the frame sample rates and frame width of feature sequences for HMM OCR system for each type independently. This type-dependent approach to compute the frame sample rate and frame width shows significant improvement in OCR accuracy over type-independent approaches. Sukhija et al. (2013) proposed a handwritten and printed recognition system using morphological operations. A set of prominent structural features is extracted to precisely distinguish one character from the other. The classification process makes use of a decision tree classifier where at each node the decision rules are defined by some morphological operations till the final realization is done. The decision trees have been optimized for performance based on classification algorithms. The results obtained are prominent and the accuracy of system is on an average 95% for handwritten text and for printed text in one font, it achieves an accuracy of 99%. A form of iterative contextual modeling that learns character models directly from the document it is trying to recognize is proposed by Kae et al. (2011). These learned models are used both to segment the characters and to recognize them in an incremental, iterative process. Results show accuracy of 98.1% in recognizing an English document in a dominant font. The speed of this iterative process is about 8–12 h.

Printed texts are not limited to one font and there are many variant fonts in texts. In these cases it is extremely difficult to identify similar characters because each font has unique shape. Therefore, it needs an OCR system that is able to recognize printed characters in many fonts. Several researches for different languages have been done to solve this problem and recognize multifont characters. Slimane et al. (2011) represented a multifont and multisize Arabic recognition system. This system is based on Hidden Markov Model Toolkit (HTK) and Bernoulli HMMs (BHMMs), that is, HMMs in which conventional Gaussian mixture density functions are replaced with Bernoulli mixture probability functions. Several tests evaluated accuracy on APTI database. The test images presented to the system are the ones rendered using a font in sizes 6, 8, 12, 18 and 24; accuracy of 98.3% is achieved. By testing 5 fonts, the accuracy is decreased about 10–20%. A system is made to develop an algorithm for recognition of machine printed isolated Kannada vowels and numerals of different font sizes and styles using modified invariant moments and they are invariant with respect to rotation, scale and translation by Hangarge Mallikarjun and Dhandra (2010). A minimum distance nearest neighbor classifier is adopted for classification. The proposed algorithm is experimented on 1800 images of vowels and 1000 images of numerals. The experimental results confirm the recognition accuracy as of 97.7% for vowels and 98.92% for numerals. A new skeletonization algorithm using modified Block Adjacency Graph (BAG) structure to recognize characters is proposed by Lakshmi et al. (2009). Computational performance on three popular fonts and sizes of an Indian script, Telugu, is tested and it is shown that the method indeed amplifies the dissimilarity between different characters and similarity between same characters in different fonts. Siriteerakul (2013) proposed and investigated the performance of a classification system that uses histogram of oriented gradient as an image feature with support vector machine as a classification tool to recognize mixed multifont Thai-English characters. The experiments were done on the datasets provided by NECTEC which consists of over 600,000 printed images of individual characters from 142 distinct classes and an accuracy of 97% can be achieved. Ben Moussa et al. (2008) proposed multilingual automatic identification of Arabic and Latin in both handwritten and printed script. This method is based on global texture analysis, by extracting fractal multidimensions features. The proposed system has been tested for 1000 prototypes with various three font types and sizes. The accuracy discrimination rate is about 96.64% by using KNN, and 98.72% by using RBF. Dhandra et al. (2008) presented an approach based on modified invariant moments for recognition of multifont English characters. The work treats isolated English characters which are normalized to a size of 33×33 pixels and the image is thinned. For size and translation invariance the modified invariant moments suggested by Palaniappan have been evaluated. The system is trained and tested for 7 different font styles with 7280 images in sizes 8–72 and success rate is found to be 99.65%. A multifont and multisize system is proposed by Rani et al. (2013). Experiments with Gabor features based on directional frequency and Gradient features based on gradient information of an individual character to identify as Gurumukhi or English are reported. 2431 trains and 4862 tests samples are in 17 various fonts in sizes 10–28 and accuracy 96.47% and 98.08% is achieved for Gradient and Gabor features, respectively.

Despite extensive studies conducted to recognize characters, existing methods have a major problem, being very sensitive to the fonts, sizes, and/or characters mode (Italic, Bold and Regular) in testing phase. This problem particularly causes higher error rate when we use the existing methods to recognize characters in different fonts rather than the training samples. Therefore, the size of the training database of existing methods is too large so that they have a

representative per font and size during the training stage. Also, the number of available fonts increases over time to satisfy different tastes. Therefore, the existing OCR methods are not able to recognize these samples in new fonts and need to change. But if there was an OCR system resistant to changes in fonts of the samples, there would not be any failure in samples recognition. This paper proposes such a system.

In this paper, first we do some preprocessing to prepare the image for extracting features. Then we use simple methods to select appropriate features from the character images. Afterwards, the extracted features are inputs of SOM neural network. This unsupervised network will classify and recognize characters in high accuracy. We have only two training samples and it does not need to have a training sample per font, so the proposed training set is small and it increases speed of recognition.

Kohonen's Self-Organizing Map (SOM) is an unsupervised learning algorithm and a powerful tool used in many areas such as data mining, analysis, classification and visualization (Tokunaga and Furukawa, 2009). As SOM is used, Euclidean distance compares instances of a class but in this paper, similarity measure is used instead of using Euclidean distance for comparing the similarity between input character's feature and various neurons' weight. One of the contours of the proposed method in this paper is that the network needs small number of samples in each class during a training phase. After the training phase, the system will be able to recognize characters, similar in sizes, fonts and different modes.

The following sections are focused in this paper: in Section 2, the database is introduced. The method and recognizing characters' steps are described in Section 3. Section 4 includes the implementation results of the proposed method. Finally the conclusions are drawn in Section 5.

2. Database details

The database used in this paper is binary images of English characters with different fonts and sizes. This database includes 237 samples of each character that are in fonts: "Times, Times New Roman, Arial, Calibri, Cambria, Arial Rounded MT Bold, Georgia, Microsoft Sans Serif, Comic, Century Schoolbook, Garamond, Verdana, Coolvetica, Courier PS, Palatino, Bookman, New Century Schoolbook, Lucida Sans Typewriter, Rockwell, Bodoni MT, Tahoma, Harry P, Copperplate Gothic Light and Felix Titling". To compare the proposed method with other methods, we made a database like them; so, the samples are selected out of first 17 fonts in sizes 10, 11, 12 and 28 (couple sizes). The samples in the next 7 fonts are in sizes 16, 18, 20, 22 and 24. Also, we added 15 bold and italic samples in fonts: "Times New Roman, Arial, Bodoni MT, Calibri, Cambria, Tahoma,[1] Rockwell and Century Gothic" in size 24 to the database. Therefore, the total number of database samples is 6162. Fig. 1 shows examples of various fonts of each character in the database. For preparing the training set, only two samples of each category have been chosen.

3. Methodology

The proposed method for recognizing or classifying characters has five steps. First, several pre-processing such as removing noise and resizing characters are done in order to do normalization. Second, the feature vector is extracted. We divide the images into two parts based on center of the images. Since the images are in binary, counting the number of ones in rows and columns of each binary image in each part of the images results in a feature vector. As stated earlier, classifying the data is done based on their similarity to each other. Third, according to the characteristics of the data feature vector, an appropriate similarity measure should be selected. Fourth, the appropriate features of high-dimension extracted feature vector are selected using genetic algorithm and SOM network is trained to determine the suitable representative for each of the different data categories. Finally, the network is evaluated based on test data.

3.1. Preprocessing

Before forming the feature vector, some pre-processing should be done on the input image. When characters are being scanned, noise is usually associated with them and a median filter should be used to remove it. After noise removal, we rotate the images 45° in the opposite direction clockwise, because it makes more difference between the

[1] There is only bold sample in font Tahoma in database.

Fonts	Characters	Fonts	Characters
Times	ABCDEFGHIJKLM NOPQRSTUVWXYZ	Calibri	ABCDEFGHIJKLMNOPQR STUVWXYZ
Times new roman	ABCDEFGHIJKLMN OPQRSTUVWXYZ	CourierPS	ABCDEFGHIJKLMNOPQ RSTUVWXYZ
Comic	ABCDEFGHIJKLMN OPQRSTUVWXYZ	ArialRoundedMTBold	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Georgia	ABCDEGHIJKLMN OPQRSTUVWXYZ	Bodoni MT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Lucida Sans Typewriter	ABCDEFGHIJKLMNO PQRSTUVWXYZ	Rockwell	ABCDEFGHIJKLMNOP QRSTUVWXYZ
Cambria	ABCDEFGHIJKLMNOPQ RSTUVWXYZ	CenturySchoolBook	ABCDEFGHIJKLMO PQRSTUVWXYZ
Harry P	ABCDEFGHIJKLMNOPQRS TUVWXYZ	Copperplate Gothic Light	ABCDEFGHIJKLM NOPQRSTUVWXY Z
Garamond	ABCDEFGHIJKLMN OPQRSTUVWXYZ	Verdana	ABCDEFGHIJKLMNO PQRSTUVWXYZ
Arial	ABCDEFGHIJKLMN OPQRSTUVWXYZ	Coolvetica	ABCDEFGHIJKLMNOPQRS TUVWXYZ
Palatino	ABCDEFGHIJKLMN OPQRSTUVWXYZ	BookMan	ABCDEFGHIJKLMN OPQRSTUVWXYZ
MicrosoftSansSerif	ABCDEFGHIJKLMN OPQRSTUVWXYZ	Tahoma	ABCDEFGHIJKLMNOPQ RSTUVWXYZ
NewCenturySchoolBo ok	ABCDEFGHIJKL MNOPQRSTUVW XYZ	Felix Titling	ABCDEFGHIJKLMNO PQRSTUVWXYZ

Fig. 1. Samples of font styles used in the study.

characters such as “I” and “J” whose top part is similar. Then we separate the character from the background for each character image by extracting the smallest rectangle surrounding each character. Since the fonts and sizes of different characters are not the same, the size of their surrounding rectangular will not be the same either. Thus, for normalizing images sizes, we resize them to 30×30 pixels similar to Maitre (1995). The resizing is done by using the nearest neighborhood interpolation method. This method is simple and without complex mathematical calculations which makes it suitable for discrete data. In addition, this method is faster than other interpolation methods, such as linear (Kahya, 2005), quadratic (Sablonnière, 1982) and cubic (Duan et al., 2000) methods (Dunlop, 1980). Fig. 2 shows an example of the preprocessing processes for character A.

3.2. Feature extraction

Extracting features is the key process and it affects the final recognition performance. Therefore the extracted feature must describe characters of each class and be able to represent a unique characteristic for each set of characters. In other words, different sizes and fonts of each of the 26 characters must have a similar feature vector so that the characters classification is performed accurately.

As discussed in Section 2, pixels of a character are displayed with value of one and background pixels with value of zero. In the proposed method, we divide the image into two parts in horizontal direction based on center of the image. In top part of the image, the total number of 1 s in each of the rows and columns is calculated in proportion of 1/3 and 2/3. In bottom part of the image, we divide it in horizontal and vertical directions and count the number of 1 s in each formed zone of the image. Each character image is stored in 30×30 pixels in the preprocessing phase and

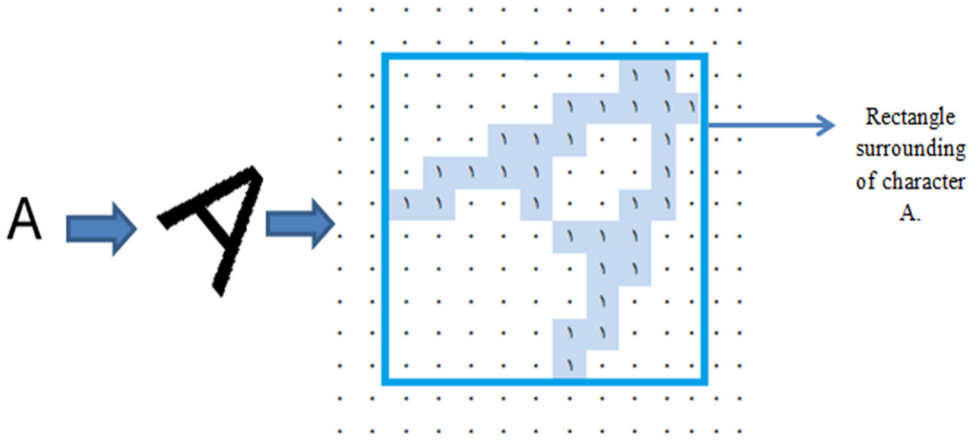


Fig. 2. A printed example of A, rotated A in 45° in the opposite clockwise and its rectangle surrounding to form feature vector.

every character is converted to a time series of length 165. The extracted features from top and bottom part of a printed pattern "A" are described in Figs. 3 and 4, respectively. Fig. 5 shows the final extracted feature of that. In Fig. 6, time series (feature vector) extracted for four different font samples of characters from A to I are displayed.

In Fig. 6, time series (feature vector) extracted for four different font samples of characters from A to I are displayed. Fig. 6 also illustrates that the feature vectors have the same length but they might not have equal amplitude changes. However, the location of local maximum or minimum can be different in the feature vectors for the different fonts of the same character. Therefore, by using a distance measure such as Euclidean distance, an accurate comparison cannot be made among different characters. While there are similarities between the feature vectors of characters in a class, it is necessary to compare and classify characters by using the similarity measure (see Fig. 6).

3.3. Similarity measure

In order to measure similarity between two time series, one similarity measure can be used. There are many similarity measures while each of them is appropriate for certain applications. Further information about similarity measures can be found in Hassanpour and Khalili (2011). In this study, Jensen similarity measure is used to measure the similarity among feature vectors of data. This measure is a useful tool for assessing the similarity of time series with the same length but with various amplitudes. Formula for comparison between two vectors of the same size P and Q based on Jensen similarity measure is expressed by (1). The output of this formula is between 0 and 1. If two sequences P and Q are identical or very similar, the output of the Jensen function will be equal to 0.

$$\frac{1}{2}\sum_{i=1}^{k}\left\{p'_i\log_2 p'_i + q'_i\log_2 q'_i - (p'_i+q'_i)\log_2\left(\frac{p'_i+q'_i}{2}\right)\right\} \quad \text{where} \quad p'_i = \frac{p_i}{\sum_{i=1}^{k} p_i} \quad \text{and} \quad q'_i = \frac{q_i}{\sum_{i=1}^{k} q_1} \tag{1}$$

The ten time series of features extracted from A to Z in ten different data sets are compared using Jensen similarity measure. Each of these time series are obtained by taking the feature vector average of ten different samples while each sample has a variety of fonts and different states. The results of this evaluation show that Jensen measure considers feature vectors extracted from a set of characters as identical (because the function will result in zero) and considers the feature vectors of two non-identical characters as dissimilar (it shows a number above 0.05, hence as threshold point).

3.4. Network structure

In this study, the neural network SOM is used for classifying characters. This network has one input layer and one output layer (Duan et al., 2000). The number of inputs is 165 which is equal to the length of the feature vector. Since each character is placed in a distinct class, this network has 26 neurons in output layer.

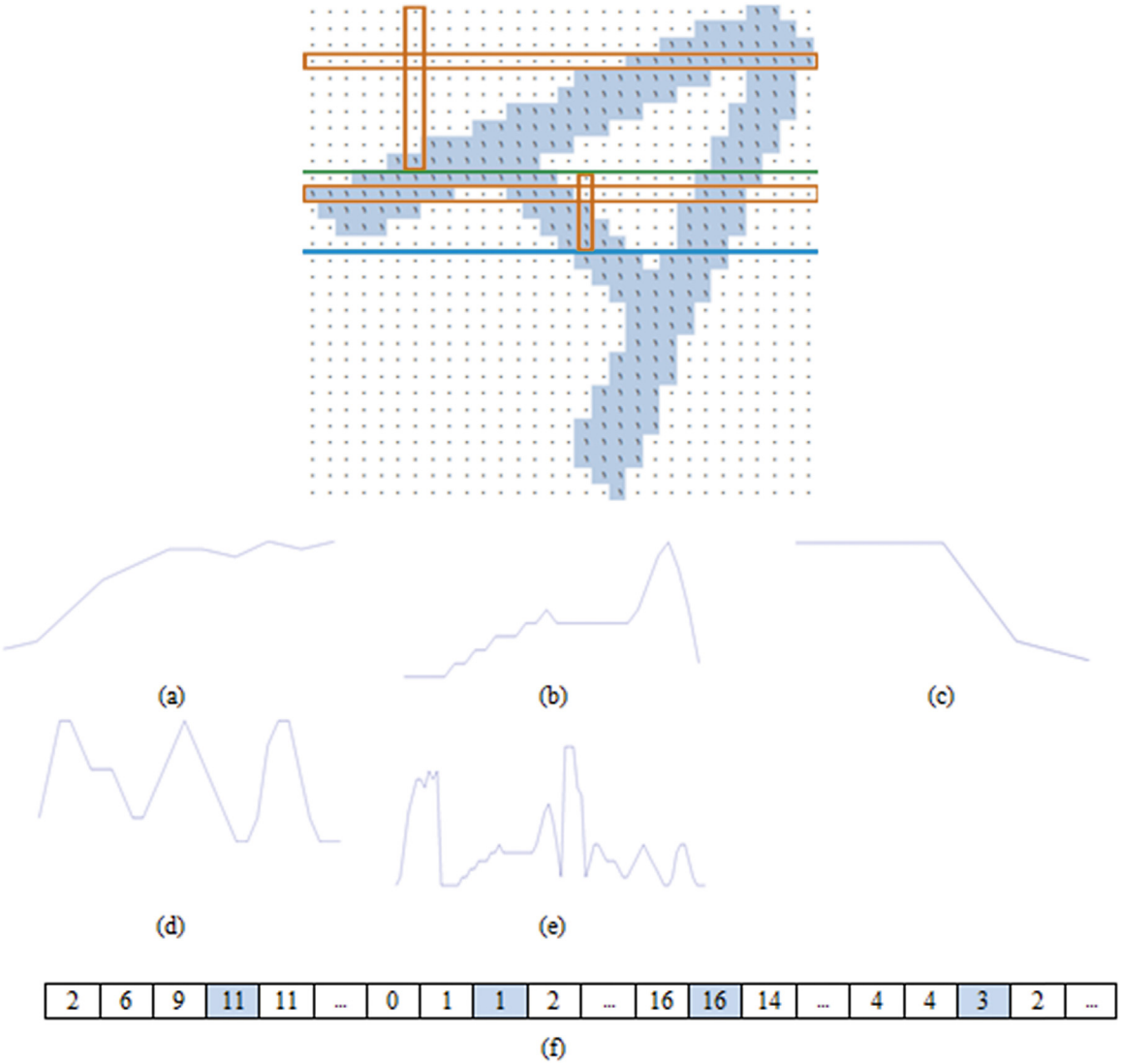


Fig. 3. Pattern A in Arial font: (a) the time series extracted of sum of 1 s in consecutive rows of 2/3 top part of image, (b) the time series extracted of sum of 1 s in consecutive columns of 2/3 top part of image, (c) the time series extracted of sum of 1 s in consecutive rows of 1/3 top part of image, (d) the time series extracted of sum of 1 s in consecutive columns of 1/3 top part of image, (e) resulting from the combined time series a, b, c and d, (f) feature vector extracted in length 75. The highlighted numbers show values of sum of specified rows and columns in image of A.

During the training stage, the network weights are initialized with a random number which is smaller than a unit. In this network, unlike conventional SOM networks using the Euclidean distance, Jensen measure is used for determining a winner neuron. Network training is repeated based on SOM networks training principle and the weight of winner neuron is updated in each iteration to determine the best representative of each character.

When the network training is finished, the weight of each of the neurons in output layer is a representative of each of the characters. By applying learning samples in the network, the most similar neuron weight in that sample is selected and the classification is done based on it.

3.5. Select features using genetic algorithm

The extracted features make a feature vector in high dimension (165 features); so it needs a manner to reduce dimension. A method for reducing dimension is removing correlation between dimensions by using methods such

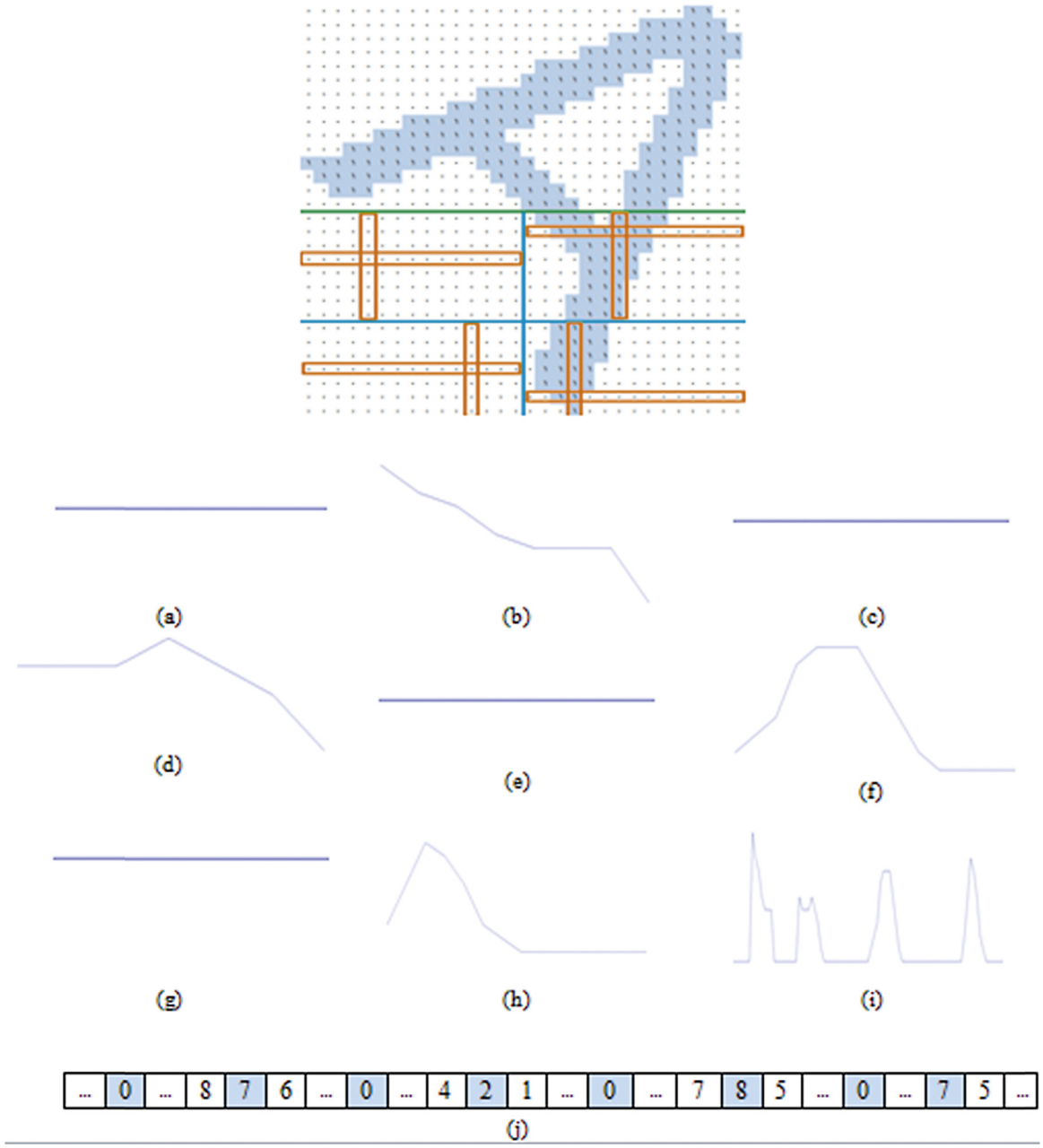
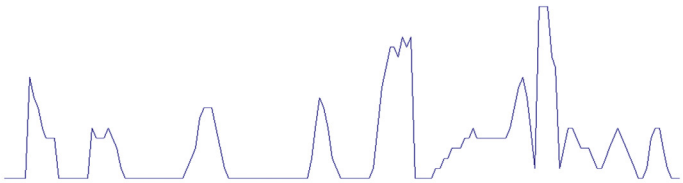


Fig. 4. Pattern A in Arial font: (a, b, c, d) the time series extracted of sum of 1 s in consecutive rows of first, second, third and fourth of bottom part of image, (e, f, g, h) the time series extracted of sum of 1 s in consecutive columns of first, second, 3rd and 4th of bottom part of image, (i) resulting from the combined time series a, b, c, d, e, f, g and h, (j) feature vector extracted in length 90. The highlighted numbers show values of sum of specified rows and columns in image of A.

as ICA and PCA that makes reduction in classifying cost. But these methods do not affect in number of extracted features from images. In this paper, we use genetic algorithm to select best features from high-dimension feature vector. Therefore, we have a reduced feature vector and fewer features.

Genetic algorithm is one of the best existing methods for optimal search (Goldberg, 1989). This algorithm is a useful tool for selecting the best features. Selecting features by genetic algorithm is presented by Siedlecki (1989). In this method, genetic algorithm is used for finding an optimal binary vector where each bit corresponds to a feature. If "i"th value of this vector is 1, "i"th feature is used in classifying and if the value is 0, corresponding feature is removed.



(a)

. . .	0	. . .	8	7	6	. . .	0	. . .	4	2	1	. . .	2	6	9	11	. . .	5	4	1	0	0	0

(b)

Fig. 5. (a) time series of pattern A in Figs. 3 and 4, (b) the final extracted feature vector of pattern A.

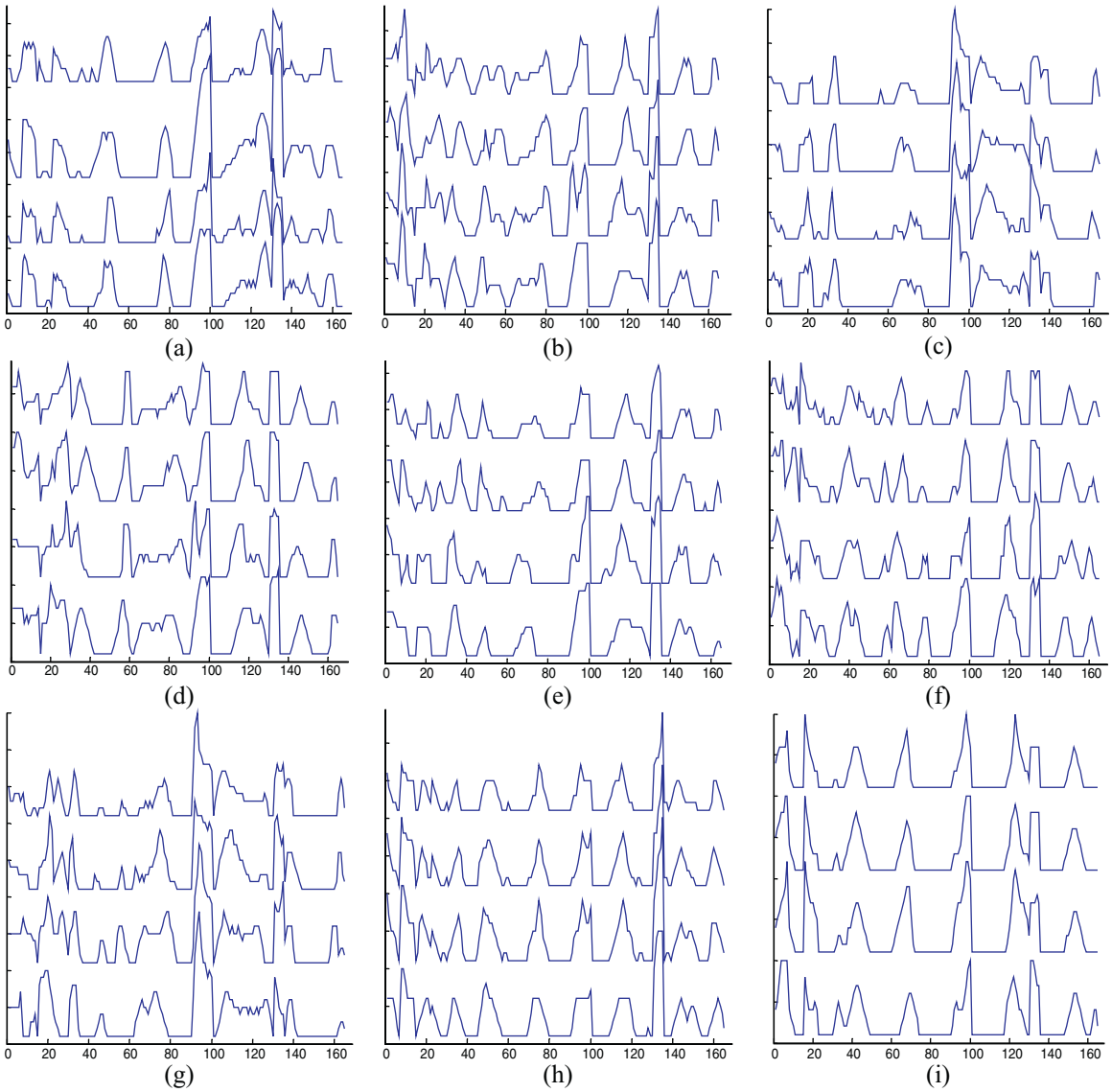


Fig. 6. The time series of feature vector extracted of characters (a) A, (b) B, (c) C, (d) D, (e) E, (f) F, (g) G, (h) H, (i) I. In each of these forms the time series of four different samples of each character are listed from top to bottom in these fonts "Lucida Sans Typewriter 'Rockwell' Bodoni MT and Palatino". The horizontal axis is the number of features.

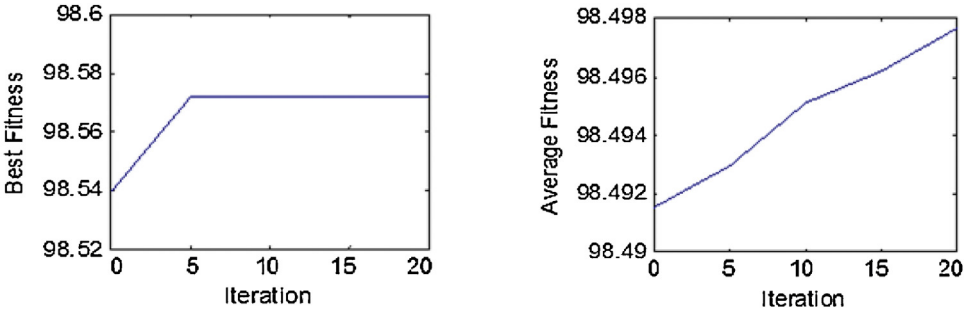


Fig. 7. The values of best and average fitness in 5 iterations of genetic algorithm.

Table 1
Different scenarios for a two-class prediction.

		Predicted classes	
		Yes	No
Actual class	Yes	True Positive	False Negative
	No	False Positive	True Negative

In a genetic algorithm, a population of candidate solutions to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; solutions are represented in binary as strings of 0 s and 1 s. The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. In this paper, by using this algorithm, we could obtain a feature vector in length 80 from feature vector in length 165. This reduction addition to reducing feature vector has increased the recognition accuracy from 98.43% to 98.56%. The objective function used is recognition accuracy and 20 chromosomes in binary generate the first generation. In each iteration, 4 chromosomes with the best fitness are selected to do mutation and cross over and are formed in the next generation. Fig. 7 shows values of the best fitness and average fitness in 20 iterations of genetic algorithm.

4. Results

To introduce a measure of accuracy and other performance measures, it should be noted that four predictions could be with the assumption of having a set of two classes of yes and no (see Table 1).

True Positive (TP) and True Negative (TN) are correct classifications. False Positive (FP) happens when a sample which is truly negative is predicted as positive. Also, False Negative (FN) happens when a positive sample is predicted as negative. Therefore, the accuracy or overall success rate is the proportion of true results (both TP and TN) in the population (2). Also for the evaluation of learner, other parameters are used such as precision, recall, and F-measure (Dunlop, 1980). Recall parameter shows what proportion of positive classes the learner predicts correctly (3). Precision or positive predictive value is defined as the proportion of the true positives to all the positive results (both true positives and false positives) (4). F-Measure considers both the precision and the recall of the test and is harmonic mean of them (5). Total F-Measure in (6) is sum of the proportion of multiplying F-measure to true positives of each class in the population.

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Table 2
The comparison between the proposed method and others.

	Number of features	Number of fonts	Resistant to character modes (bold, italic)	Classifying method	Number of training and test samples	Accuracy (%)
Rani et al. (2013)	189	17	×	Support vector machine	2431 trains – 4862 tests	96.47
Rani et al. (2013)	200	17	×	Support vector machine	2431 trains – 4862 tests	98.08
Proposed method	80	17	√	SOM neural network	52 trains – 4862 tests	98.68
Proposed method	80	24	√	SOM neural network	52 trains – 6162 tests	98.56

Table 3
The accuracy of the proposed method on database.

	PR	R	FM		PR	R	FM		PR	R	FM		PR	R	FM
A	1	0.99	0.99	H	0.89	0.98	0.93	O	0.97	0.98	0.98	V	1	0.98	0.99
B	0.99	0.97	0.98	I	0.98	1	0.99	P	0.99	0.98	0.99	W	1	1	1
C	1	0.98	0.99	J	1	0.99	0.99	Q	0.99	0.98	0.98	X	1	1	1
D	0.99	0.99	0.99	K	0.99	0.99	0.99	R	0.95	0.98	0.96	Y	0.98	0.99	0.99
E	0.97	0.99	0.98	L	1	0.99	0.99	S	1	0.99	0.99	Z	0.99	1	0.99
F	1	1	1	M	0.99	0.90	0.94	T	0.99	1	0.99				
G	0.98	0.99	0.98	N	1	0.94	0.97	U	0.96	1	0.98				

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$PR = \frac{TP}{TP + FP} \tag{4}$$

$$FM = 2 \times \frac{PR \times R}{PR + R} \tag{5}$$

$$TFM = \sum \frac{TP}{n} \times (\mathrm{FM}) \quad n \text{ is all data} \tag{6}$$

Two tests are performed to evaluate performance of the proposed method. In the first experiment, we compare the proposed method to another method by the database similar to it. This database includes binary samples in 17 fonts and 11 sizes introduced in Section 2. Table 2 shows the comparison between the proposed method and others. The proposed method has higher accuracy than others. We added the samples in more fonts to do the second experiment. In this experiment, the samples are in 24 fonts, bold and italic. By applying the proposed method, the characters presented in the introduced database are recognized and classified. As Table 2 shows, when we add samples in more fonts, the accuracy has not much decreased; it emphasizes the proposed method is resistant to changing number of fonts; because by adding samples in more fonts to the database, the accuracy stays constant and it has not rapidly decreased such as other OCR methods. So, we could be confident that the proposed method will be successful in recognizing characters even if the number of fonts increases more than 24. Also, the number of training samples has not been changed when the number of fonts added to database is increased; it means we do not need to add a new sample in new font to training set and the proposed method could recognize the characters based on the previous training samples. However, other methods have to add a sample of new font to training set when they add samples with new fonts to their database. As seen in the Table 2, requiring small number of training samples, being resistant to font variant, being resistant to character modes, simple classification method and high accuracy are advantages of the proposed method compared with the other methods.

Table 3 shows performance of proposed method based on formulas (3, 4, and 5). Values of 1 of PR in Table 3 show the proposed method has been truly classified characters; however the other values of PR illustrate which characters

Table 4
Accuracy of the proposed method for each digit.

	Accuracy (%)		Accuracy (%)		Accuracy (%)		Accuracy (%)
A	99	H	98.31	O	98	V	98
B	97.47	I	100	P	98	W	100
C	98.31	J	99	Q	98	X	100
D	99.16	K	99	R	98	Y	99.57
E	99	L	99	S	99	Z	100
F	100	M	90	T	100		
G	99.16	N	94.09	U	100		

have been selected as wrong characters. The similarity between M and H feature vectors makes wrong classifying in these two categories. It shows the need for more accurate features for distinguishing these similar characters.

Table 4 shows the accuracy in classing each of the characters. In order to evaluate the proposed method more accurately, AC and TFM values of classifying data in the experiment are 98.56% and 0.97%, respectively.

5. Conclusions

The purpose of this study is to represent a method which is able to identify different characters with different fonts. The proposed method could recognize the characters with 98.56% accuracy by using a simple neural network without using a large training database. In this method, data are initially rotated and normalized in the preprocessing step. Then, the image is divided into two parts in horizontal direction and the features are extracted by whisking data in the direction of row and column in each part of image. These features are similar for a character even with different fonts. SOM neural network is used for recognizing and classifying characters in this paper, and a similarity measure is employed to identify the class of data in a neural network. The evaluations in this study show that the proposed method is able to recognize English characters in different fonts with high accuracy after training. Comparing with other methods in the literature shows that requiring small number of training samples, using a simple classification approach, and being resistant to font variant are advantages of the proposed method.

References

Ben Moussa, S., Zahour, A., Benabdelhafid, A., Alimi, A.M., 2008. Fractal-based system for Arabic/Latin, printed/handwritten script identification. In: 19th International Conference on Pattern Recognition, 2008. ICPR 2008, 8–11 December 2008, pp. 1–4, http://dx.doi.org/10.1109/ICPR.2008.4761838.

Dhandra, B.V., Mallikarjun, V.S.M., Ravindra Hegadi, H., 2008. Multi-font English character recognition based on modified invariant moments. J. Comb. Math. Comb. Comput. 76, 153–162.

Duan, Q.K.D., Price, W.G., Twizell, E.H., 2000. Weighted rational cubic spline interpolation and its application. J. Comput. Appl. Math. 117 (2), 121–135.

Dunlop, G.R., 1980. A rapid computational method for improvements to nearest neighbor interpolation. Comput. Math. Appl. 6 (3), 349–353.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley.

Hangarge Mallikarjun, P.S., Dhandra, B.V., 2010. Multi-font/size Kannada vowels and numerals recognition based on modified invariant moments. Int. J. Comput. Appl. (Special issue on RTIPPR (2)), 126–130.

Hassanpour, H.A.D., Khalili, A., 2011. A regression-based approach for measuring similarity in discrete signals. Int. J. Electron., 98.

Huaigu, C., Prasad, R., Natarajan, P., 2011. Handwritten and typewritten text identification and recognition using hidden Markov models. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), 18–21 September 2011, pp. 744–748, http://dx.doi.org/10.1109/ICDAR.2011.155.

Kae, A., Smith, D., Learned-Miller, E., 2011. Learning on the fly: a font-free approach toward multilingual OCR. Int. J. Doc. Anal. Recognit. (IJDAR) 14 (3), 289–301, http://dx.doi.org/10.1007/s10032-011-0164-6.

Kahya, E., 2005. A new unidimensional search method for optimization: linear interpolation method. Appl. Math. Comput. 171 (2), 912–926.

Lakshmi, C.V., Singh, S., Jain, R., Patvardhan, C., 2009. A novel approach to skeletonization for multi-font OCR applications. In: Chaudhury, S., Mitra, S., Murthy, C.A., Sastry, P.S., Pal, S. (Eds.), Pattern Recognition and Machine Intelligence. Lecture Notes in Computer Science, vol. 5909. Springer, Berlin, Heidelberg, pp. 393–399, http://dx.doi.org/10.1007/978-3-642-11164-8_64.

Maitre, G., 1995, June. Experiments with robust similarity measures for OCR, IDIAP TR 95-103.

Rahman, A.F.R., Fairhurst, M.C., 1998. Machine-printed character recognition revisited: re-application of recent advances in handwritten character recognition research. Image Vis. Comput. 16 (12–13), 819–842, http://dx.doi.org/10.1016/S0262-8856(98)00056-0.

Rani, R., Dhir, R., Lehal, G.S., 2013. Script identification of pre-segmented multi-font characters and digits. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), 25–28 August 2013, pp. 1150–1154, http://dx.doi.org/10.1109/ICDAR.2013.233.

Sablonnière, P., 1982. Interpolation by quadratic splines on triangles and squares. Comput. Ind. 3 (1–2), 45–52.

Siedlecki, J.S., 1989. A note on genetic algorithm for large scale feature selection. Pattern Recognit. Lett. 10, 335–347.

Siriteerakul, T., 2013. Mixed Thai-English character classification based on histogram of oriented gradient feature. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), 25–28 August 2013, pp. 847–851, http://dx.doi.org/10.1109/ICDAR. 2013.173.

Slimane, F., Kanoun, S., Abed, H.E., Alimi, A.M., Ingold, R., Hennebert, J., 2011. ICDAR 2011 – Arabic recognition competition: multi-font multi-size digitally represented text. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), 18–21 September 2011, pp. 1449–1453, http://dx.doi.org/10.1109/ICDAR.2011.288.

Sukhija, S., Panwar, S., Nain, N., 2013. CRAMM: character recognition aided by mathematical morphology. In: 2013 IEEE Second International Conference on Image Information Processing (ICIIP), 9–11 December 2013, pp. 323–328, http://dx.doi.org/10.1109/ICIIP.2013.6707608.

Tokunaga, K., Furukawa, T., 2009. Modular network SOM. Neural Netw. 22 (1), 82–90, http://dx.doi.org/10.1016/j.neunet.2008.10.006.

Yang, Y., Lijia, X., Chen, C., 2011. English character recognition based on feature combination. Procedia Eng. 24 (0), 159–164, http://dx.doi.org/10.1016/j.proeng.2011.11.2619.